Strukturat Perseritese

Ne gjuhen programuese C gjenden tre struktura qe kryejne perseritjen e nje pjese programi ne C, te cilat jane: **For; While; Do-while.**

1. **Struktura perseritese for**

Fillon me fjalen celes **for** dhe brenda kllapave rrumbullake te saj deklarohen:

i. inicializimi i variablit te kontrollit
ii. kushti llogjik (ku vleresohet variabli i kontrollit)
iii. ndryshimi I variablit te kontrollit

Te treja pjeset ndahen nga njera tjetra me pikepresje ;. Ne pjesen e pare inicializimi I variablit te kontrollit caktohet vlera fillestare e variablit ose ekzekutohet ndonje funksion ne fillim te perseritjes. Gjithashtu kjo pjese ekzekutohet vetem nje here gjate hyrjes ne strukturen perseritese FOR. Ne pjesen e dyte nese kushti llogjik eshte I vertete percakton ekzekutimin e bllokut te veprimeve te struktures FOR, perndryshe nuk ekzekutohet asnje veprim. Ndryshimi I variablit te kontrollit ekzekutohet pas cdo perfundimi te ekzekutimit te bllokut te veprimeve te struktures FOR.

Ajo ka formen standarte si me poshte:

```
for( inicializimi I variablit kontrollit; kushti llogjik; ndryshimi I variablit te kontrollit)
{
  veprime;
}
```

Me poshte jane paraqitur disa shembuj me strukturen perseritese for:

/* llogarit interesin e thjeshte per 3 bashkesi p, n dhe r te dhena nga perdoruesi:*/

```
#include <stdio.h>
 main( )
 {
 int p,n, i;
 float r, si;
 for(i = 1; i <=3; i++)
 {
 printf("Enter values of p,n and r");
 scanf("%d %d %f", & p, & n, & r);
 si = p * n * r / 100;
 printf("Simple interest = Rs %f ", si);
 }
 }
#include <stdio.h>
main()
{
  int ans=0;  int i;
   for (i=0;i< 10; i++)
   {   ans += i;
       printf("i= %d   ",i);
      printf("ans= %d\n", ans);
   }
}
```

Si rezultat i ekzekutimit te struktures perseritese te mesiperm do te kishim nje tabele me vlera te i dhe ans si me poshte:

i=0 ans=0
i=1 ans=1
i=2 ans=3
i=3 ans=6
……

i=9 ans=45

2. Strukture perseritese while

Struktura perseritese while vazhdon ekzekutimin derisa kushti llogjik eshte i vertete. Nese blloku i veprimeve perbehet me shume se nje veprim atehere perdorim { }. Nese nuk jemi te kujdesshem gjate shkruarjes se programit ku kushti llogjik nuk behet asnjehere i gabuar , atehere ekzekutimi i struktures perseritese nuk perfundon (do te ekzekutohet pa kufi here). E njejta gje ndodh edhe me strukturat FOR dhe DO..While. Pa perdorimin e strukturave perseritese programi eshte me i gjate dhe me pak i lexueshem, edhe pse kryen te njejten detyre sikurse programi qe perdor strukturat perseritese. Psh : Programi per te nxjerre ne ekran numrat nga 1 deri ne 10, si dhe vleren katrore dhe kubike te ketyre numrave pa perdorur strukturat perseritese do te ishte :

```
printf( ”numri 1   katrori %d          kubi %d”, 1*1, 1*1*1) ;
printf( ”numri 2   katrori %d          kubi %d”, 2*2, 2*2*2) ;
.....
printf( ”numri 10 katrori %d          kubi %d”, 10*10, 10*10*10) ;
```

Duke perdorur strukturat perseritese do kishim:

```
#include <stdio.h>
main()
{
int nr;
printf("Numri Katrori      Kubi\n");
nr=1;
while (nr<11)
   {
   printf("%3d %7d %7d\n", nr, nr*nr, nr*nr*nr );
   nr++;
   }
}
```

Paraqitja analitike e strukture perseritese while ne pergjithesi eshte:

```
Inicilizimi I variablit te kontrollit;
while (kushti llogjik)
{
        veprime;
        ndryshimi I variablit te kontrollit;
}
```

Programi I meposhtem llogarit interesin e thjeshte per 3 bashkesi p, n dhe r te dhena nga perdoruesi:

```
#include <stdio.h>
 main( )
 {
        int p,n, i;
         float r, si;
         i = 1;
         while(i < = 3 )
         {
                printf("Enter values of p, n and r");
                scanf("%d %d %f", & p, & n, & r);
                si = p*n*r/100;
                printf("simple interest = Rs. %f ", si);
                i++;
         }
 }
```

Paraqesim programin e mesiperm me Strukture perseritesen While dhe marrim te njejtin rezultat.

```
#include <stdio.h>
main()
{
  int ans=0; int i=0;
              while (i< 10)
              {
               ans += i;
               printf("     i= %d   ",i);
               printf("ans= %d\n",ans);
                i++;
               }
}
```

3. Strukture perseritese DO While

Struktura perseritese do-while eshte e ngjashme me strukturen perseritese while dhe ekzekutohet derisa kushti eshte I vertete. Paraqitja analitike e tij eshte si me poshte:

```
Do {
  Veprime;
    }
while(kushti llogjik)
```

Ndryshimi kryesor eshte qe vleresimi I kushtit llogjik behet ne fund mbasi ekzekutohet nje here trupi I struktures perseritese. Programin e mesiperm e shohim edhe me Strukture perseritesen do-while.

```
#include <stdio.h>
main()
{
  int ans=0; int i=0;
   do {
        ans += i;
        printf("     i= %d   ",i);
        printf("ans= %d\n",ans);
        i++;
       } while (i<10);
}
```

Instruksioni Break

Ndonjehere nevojitet qe ta kontrollojme perseritjen e strukturave perseritese te permendura me siper, jo vetem me testimin e kushtit ne fillim te cdo strukture. Fjala celes break ben daljen e menjehershme nga cikli pa pritur exekutimin e kushtit llogjik dhe kalon direkt ne veprimin e pare pas fundit te ciklit. Programi I meposhtem percakton nese numrat e dhene nga tastiera jane numra prim ose jo prim.

```
#include <stdio.h>
main( )
 {
              int num, i;
               printf("Jep nje numer");
               scanf("%d", &num);
               i = 2;
               while (i <= num -1)
```

```
        {
                if (num%i== 0)
                {
                        printf("Nuk eshte numer prim");
                        break;
                }
        i ++;
        }
        if(i == num)
        printf("Eshte numer prim");
}
```

Instruksioni Continue

Eshte I njejte me instruksionin Break. Me perdorimin e fjales celes continue nuk dalim nga struktura perseritese, por nderpresim ekzekutimin e pjeses se programit qe pason kete instruksion. Psh:

```
main( )
{
        int i,j;
        for(i = 1; i< = 2; i++)
        {
                for(j=1; j<=2; j++)
                {
                        if (i= =j)
                        continue;
                        printf("%d%d", i,j);
                }
        }
}
```

Ne ekran mbas ekzekutimin te programit te mesiperm do te kishim: 12 21

Kur vlera e I eshte e barabarte me j instruksioni continue merr nen kontroll ekzekutimin e struktures perseritese duke bere nderprerjen e veprimeve te mbetura brenda tij. Instruksioni continue perdoret zakonisht kur pjesa e ciklit qe vjen pas saj eshte shume e komplikuar dhe perdorimi I saj mundeson qe programi te jete me kompakt.

Operatori ternar ?

Instruksioni I kushtezuar me trajten e meposhtme:

```
If(nr= =1000)
        nr=1;
else
        nr++;
```

jane shume te shpeshta dhe mund te zevendesohen me operatorin ternar ?, i cili perbehet nga tri pjese: kushti, pjesa e sakte dhe e pasakte.

Psh.

```
Kushti? Pjesa e sakte : pjesa e pasakte;
```

Kodi i mesiperm i cili perdor instruksionin e kushtezuar if....else do te paraqitej me operatorin ternar:

```
Nr = (nr= = 1000) ? 1: nr++;
```

Ne instruksionin e kushtezuar se pari shqyrtohet kushti llogjik dhe nese eshte i vertete ekzekuton nr=1 dhe nese jo nr++;

Operatori ternar ? perdoret kur blloku i veprimeve ka vetem nje veprim, nese ka me shume eshte mire te perdoret instruksioni i kushtezuar if...else.